



Programming Concepts

PRACTISE SESSION #2

INTRODUCTION TO PYTHON

×

×

×

×

01

Agenda

- чому пайтон
- синтаксис мови, теорія
- практичні завдання
- розбір 2 есайменту
- захист робіт

Чому пайтон

- проста для вивчення та розуміння мова

Для чого використовується

- Для аналізу даних
- Для візуалізації даних
- Для машинного навчання
- Для розробки програмного забезпечення & веброзробки



Чому пайтон

програма - набір команд, інструкцій

```
1 Pick up catalog
2 Open catalog at the middle
3 Look at card
4 If person is on card
5     File Exist!
6 Else if person is earlier in catalog
7     Open to middle of bottom half of catalog
8     Go back to line 3
9 Else if person is later in catalog
10    Open to middle of upper half of catalog
11    Go back to line 3
12 Else
13    Quit
```



Python

```
def binary_search(catalog, person):
    low = 0
    high = len(catalog) - 1

    while low <= high:
        middle = (low + high) // 2
        if catalog[middle] == person:
            return True # The person is found
        elif catalog[middle] < person:
            low = middle + 1
        else:
            high = middle - 1
    return False

sample_catalog = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
search_target = 5
found = binary_search(sample_catalog, search_target)
found
```

Assembler

```
START:
    mov eax, 0                ; Set low to 0
    mov ebx, size             ; Set up to size of the catalog


SEARCH_LOOP:
    cmp eax, ebx
    jge END                   ; If low is not smaller than up, we are done

    mov ecx, eax
    add ecx, ebx
    shr ecx, 1                ; Calculate middle as (low + up) / 2

    ; Assume we have a function that compares the value at 'catalog[middle]'
    ; and returns if it's equal, less, or greater
    push ecx
    call CompareFunction
    pop ecx

    cmp CompareResult, 0
    je FILE_EXISTS            ; If the person is at 'catalog[middle]', we've

    cmp CompareResult, 1
    jl LOWER_HALF
    jmp UPPER_HALF
```



× × × ×

Variables (змінні)

```
# Variables  
number = 123  
message = 'Hello, fellows'
```

Task: створити дві змінні та записати їм



×

Властивості змінних

```
number_1 = 123  
number_1 = 345  
print(number_1)
```

Task: створити змінну, задати їй спершу одне значення, потім інше. Це також завітнувати



Правила називання змінних

- Variable names can contain only letters, numbers, and underscores. They can start with a letter or an underscore, but not with a number.

For instance, you can call a variable `message_1` but not `1_message`.

- Spaces are not allowed in variable names, but underscores can be used to separate words in variable names.

For example, `greeting_message` works, but `greeting message` will cause errors.

- Avoid using Python keywords and function names as variable names that is, do not use words that Python has reserved for a particular programmatic purpose, such as the word `print`.
- Variable names should be short but descriptive. For example, `name` is better than `n`, `student_name` is better than `s_n`, and `name_length` is better than `length_of_persons_name`.
- Be careful when using the lowercase letter `l` and the uppercase letter `O` because they could be confused with the numbers `1` and `0`.



× × × ×

Властивості змінних

```
name = input("Enter you duck's name")  
print('Your duck`s name is', name)  
print(f'Your duck`s name is {name}')
```

**Task: Задати значення змінної через input та
запінтувати це значення**



Типи даних (основні)

- **int** - or integer, is a whole number, positive or negative, without decimals, of unlimited length. Example, 123.
 - To get from string use: **int("123")**
- **float** - or "floating point number" is a number, positive or negative, containing one or more decimals.. Example, 123.823.
 - To get from string use: **float("123.823")**
- **string** - text values. Example "This is a string value".
 - To convert something to string use: **str(123)**
- To check data type use **type()**

Task: задати значення змінній **number** через input (число)

- перевірити тип **number** та завітнувати
- змінити тип змінної на **int** Та завітнувати тип
- змінити тип змінної на **float** та завітнувати тип



x x x x

Математичні операції

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

str + int = float

× × × ×

Час практики))

Tasks for 7 minutes:

Choose at least one:

- Get a number from user and print its square root to the console
- Get area width and length of the rectangle then return its area



× × × ×

Час практики))

Tasks for 7 minutes:

Choose at least one:

- Get a number from user and print its square root to the console
- Get area width and length of the rectangle then return its area



× × × ×

If & else operators

if cond1:

✓ ...

elif cond2:

✓ ...

else:

✓ ...

tab



× × × ×

If & else operators

if not cond1:

✓ ...

elif cond1 or cond2 and cond3:

✓ ...

else:

✓ ...



If & else operators

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

Code:

```
a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
```

Напишіть простенький калькулятор який
підтримує такі дії: + - / * та приймає 2 числа.

Алгоритм

- > інпут першого числа
- > інпут другого числа
- > інпут операції
- > if else + print result

× × × ×

Есaiймент 2



